US006898690B2

(12) United States Patent
Ramagopal et al.

(10) **Patent No.: US 6,898,690 B2**
(45) **Date of Patent: *May 24, 2005**

(54) **MULTI-TIERED MEMORY BANK HAVING DIFFERENT DATA BUFFER SIZES WITH A PROGRAMMABLE BANK SELECT**

(75) Inventors: **Hebbalalu S. Ramagopal**, Austin, TX (US); **Michael Allen**, Austin, TX (US); **Jose Fridman**, Brookline, MA (US); **Marc Hoffman**, Mansfield, MA (US)

(73) Assignees: **Intel Corporation**, Santa Clara, CA (US); **Analog Devices, Inc.**, Norwood, MA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/639,020**

(22) Filed: **Aug. 11, 2003**

(65) **Prior Publication Data**

US 2004/0034739 A1 Feb. 19, 2004

Related U.S. Application Data

(63) Continuation of application No. 09/541,114, filed on Mar. 31, 2000, now Pat. No. 6,606,684.

(51) **Int. Cl.**[7] **G06F 12/10**

(52) **U.S. Cl.** **711/202**; 711/118; 711/120; 711/168; 711/170; 711/209

(58) **Field of Search** 711/202, 118, 711/120, 168, 170, 209

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,623,990 A	11/1986	Allen et al.	365/189.04
5,001,671 A	3/1991	Koo et al.	365/230.05
5,175,841 A	12/1992	Magar et al.	711/211
5,257,359 A	10/1993	Blasco et al.	711/138
5,410,669 A	4/1995	Biggs et al.	711/118
5,465,344 A	11/1995	Hirai et al.	711/131
5,535,359 A	7/1996	Hata et al.	711/119
5,537,576 A	7/1996	Perets et al.	711/150
5,559,986 A	9/1996	Alpert et al.	711/144
5,611,075 A	3/1997	Garde	711/153
5,737,564 A	4/1998	Shah	711/202
6,023,466 A	2/2000	Luijten et al.	370/395.31
6,038,630 A	3/2000	Foster et al.	710/317
6,038,647 A	3/2000	Shimizu	711/168
6,127,843 A	10/2000	Agrawal et al.	326/40
6,189,073 B1	2/2001	Pawlowski	711/120
6,267,720 B1	7/2001	Knox et al.	712/23
6,321,318 B1	11/2001	Baltz et al.	711/170
6,334,175 B1	12/2001	Chih	711/170
6,446,181 B1	9/2002	Ramagopal et al.	711/168

FOREIGN PATENT DOCUMENTS

DE	19809640	9/1999	G06F/15/80
WO	WO98/13763	4/1998	G06F/12/08
WO	WO-99/45474	10/1999	G06F/15/00

OTHER PUBLICATIONS

International Search Report for International Application PCT/US01/10573 dated Nov. 22, 2001.

Su, et al., "A Study of Cache Hashing Functions for Symbolic Applications in Micro–parallel Processors," pp. 530–535, IEEE, 1994.

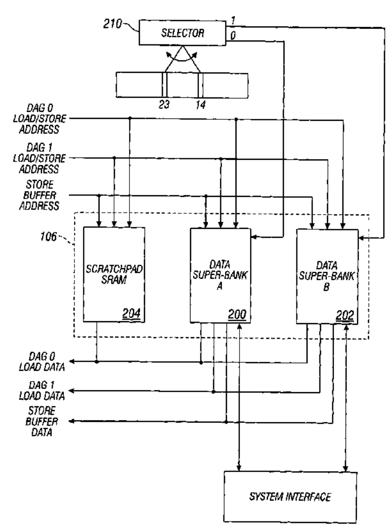
Zhang, et al., "Multi–Column Implementations for Cache Associativity," pp. 504–509, IEEE, 1997.

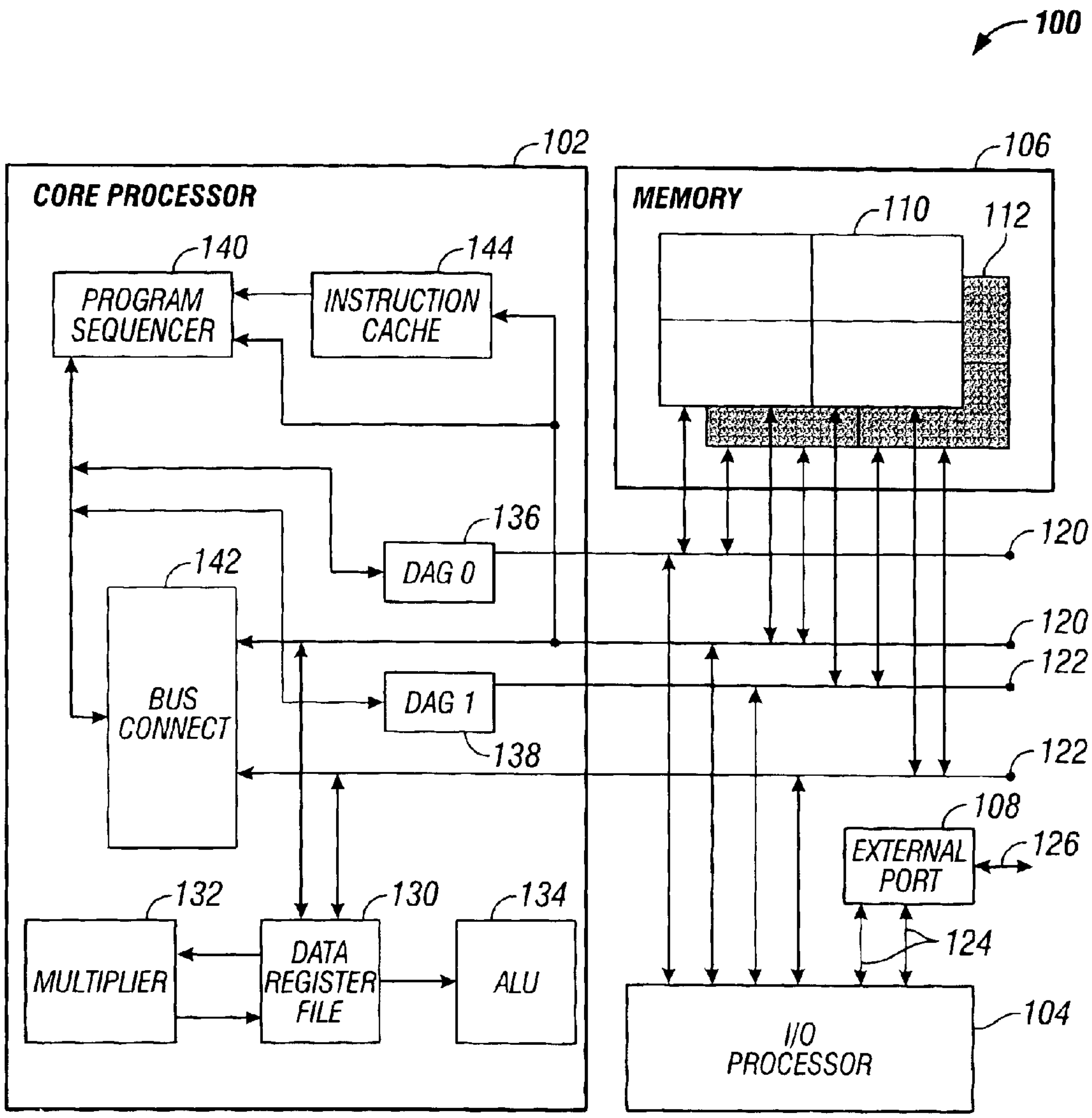
Primary Examiner—Stephen Elmore
(74) *Attorney, Agent, or Firm*—Fish & Richardson P.C.

(57) **ABSTRACT**

An apparatus having a core processor and a plurality of cache memory banks is disclosed. The cache memory banks are connected to the core processor in such a way as to provide substantially simultaneous data accesses for said core processor.

6 Claims, 6 Drawing Sheets





100
102
CORE PROCESSOR
140
PROGRAM SEQUENCER
144
INSTRUCTION CACHE
136
DAG 0
142
BUS CONNECT
DAG 1
138
132
MULTIPLIER
130
DATA REGISTER FILE
134
ALU
106
MEMORY
110
112
120
120
122
122
108
EXTERNAL PORT
126
124
104
I/O PROCESSOR

FIG. 1

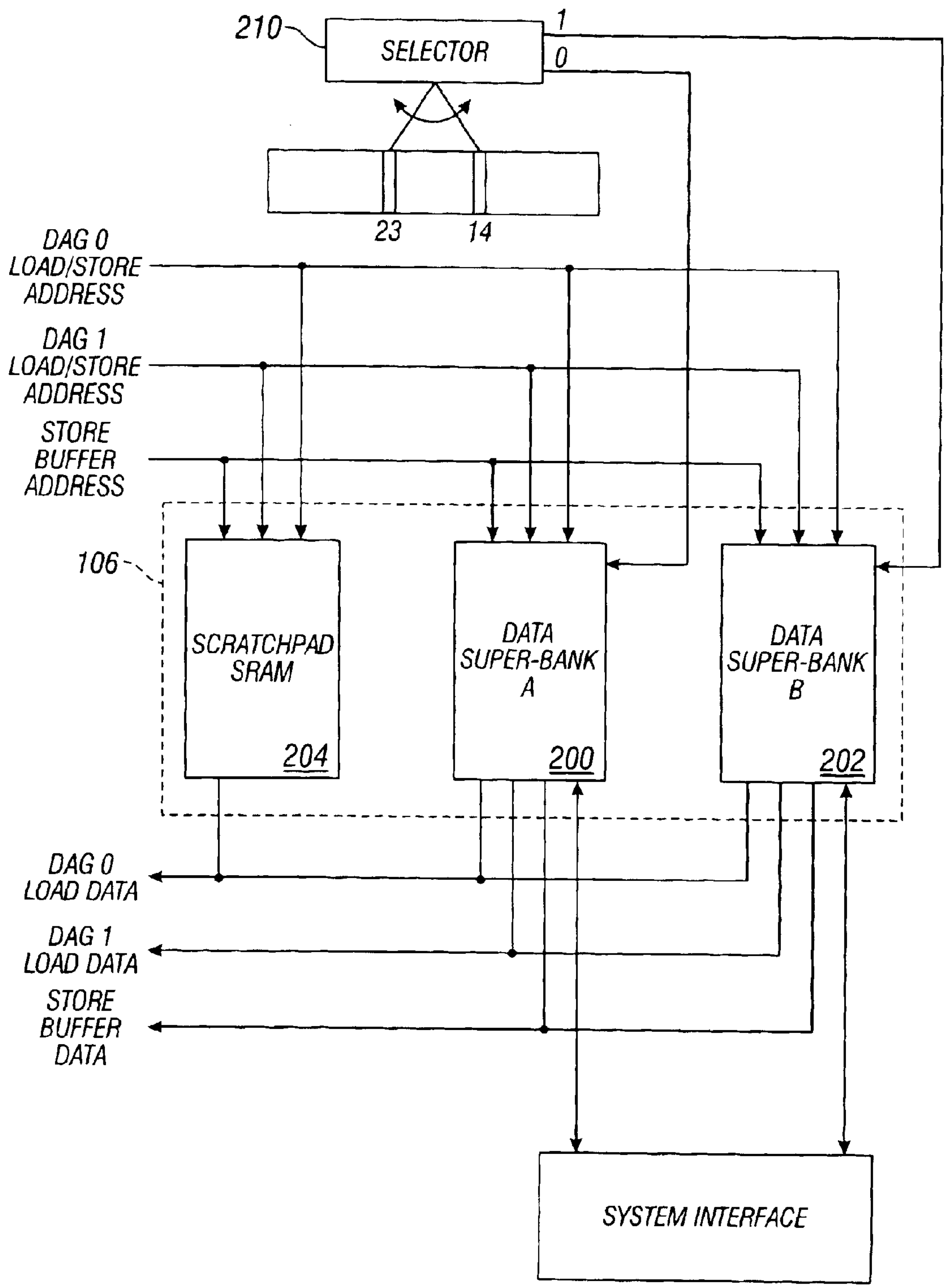

210
SELECTOR
1
0
23
14
DAG 0 LOAD/STORE ADDRESS
DAG 1 LOAD/STORE ADDRESS
STORE BUFFER ADDRESS
106
SCRATCHPAD SRAM
204
DATA SUPER-BANK A
200
DATA SUPER-BANK B
202
DAG 0 LOAD DATA
DAG 1 LOAD DATA
STORE BUFFER DATA
SYSTEM INTERFACE

FIG. 2

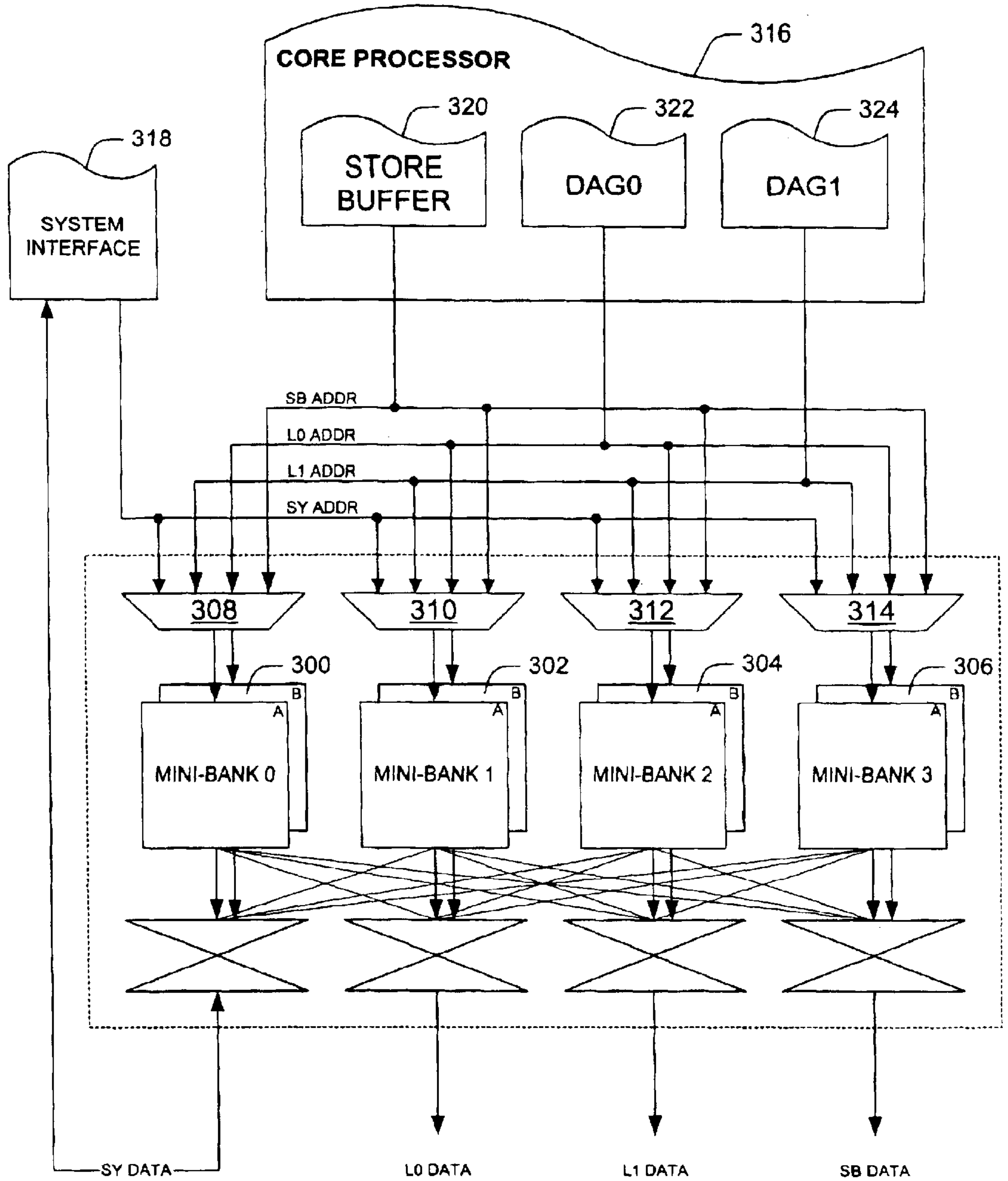

316
CORE PROCESSOR
320
STORE BUFFER
322
DAG0
324
DAG1
318
SYSTEM INTERFACE
SB ADDR
L0 ADDR
L1 ADDR
SY ADDR
308
310
312
314
300
302
304
306
B
A
MINI-BANK 0
MINI-BANK 1
MINI-BANK 2
MINI-BANK 3
SY DATA
L0 DATA
L1 DATA
SB DATA

FIG. 3

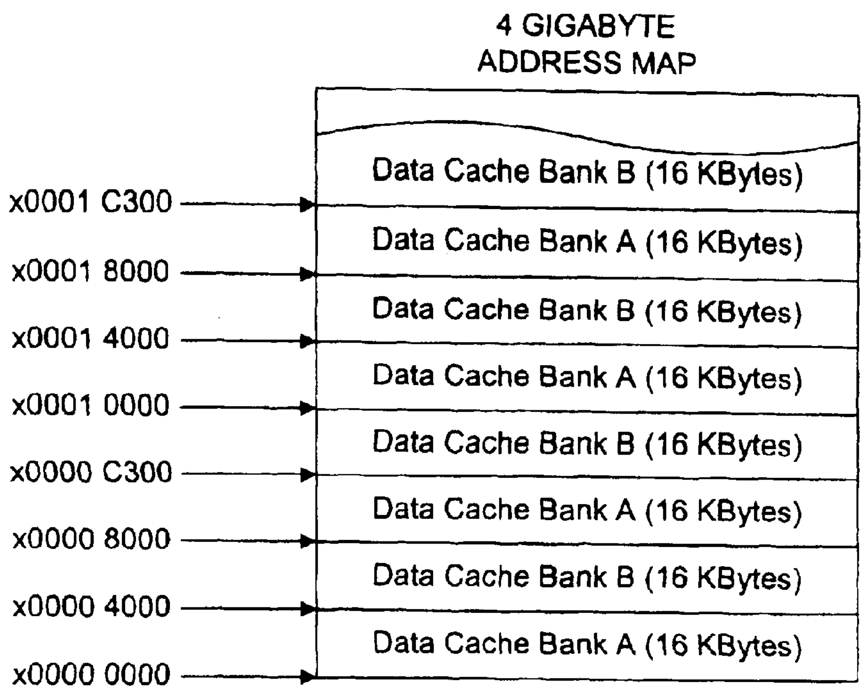

4 GIGABYTE
ADDRESS MAP
Data Cache Bank B (16 KBytes)
x0001 C300
Data Cache Bank A (16 KBytes)
x0001 8000
Data Cache Bank B (16 KBytes)
x0001 4000
Data Cache Bank A (16 KBytes)
x0001 0000
Data Cache Bank B (16 KBytes)
x0000 C300
Data Cache Bank A (16 KBytes)
x0000 8000
Data Cache Bank B (16 KBytes)
x0000 4000
Data Cache Bank A (16 KBytes)
x0000 0000

FIG. 4

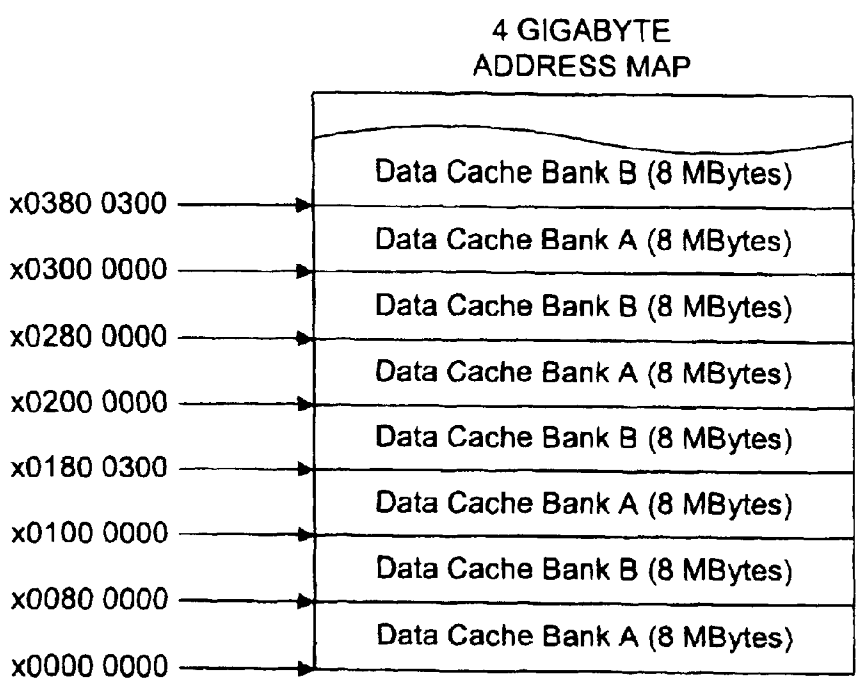

4 GIGABYTE
ADDRESS MAP
Data Cache Bank B (8 MBytes)
x0380 0300
Data Cache Bank A (8 MBytes)
x0300 0000
Data Cache Bank B (8 MBytes)
x0280 0000
Data Cache Bank A (8 MBytes)
x0200 0000
Data Cache Bank B (8 MBytes)
x0180 0300
Data Cache Bank A (8 MBytes)
x0100 0000
Data Cache Bank B (8 MBytes)
x0080 0000
Data Cache Bank A (8 MBytes)
x0000 0000

FIG. 5

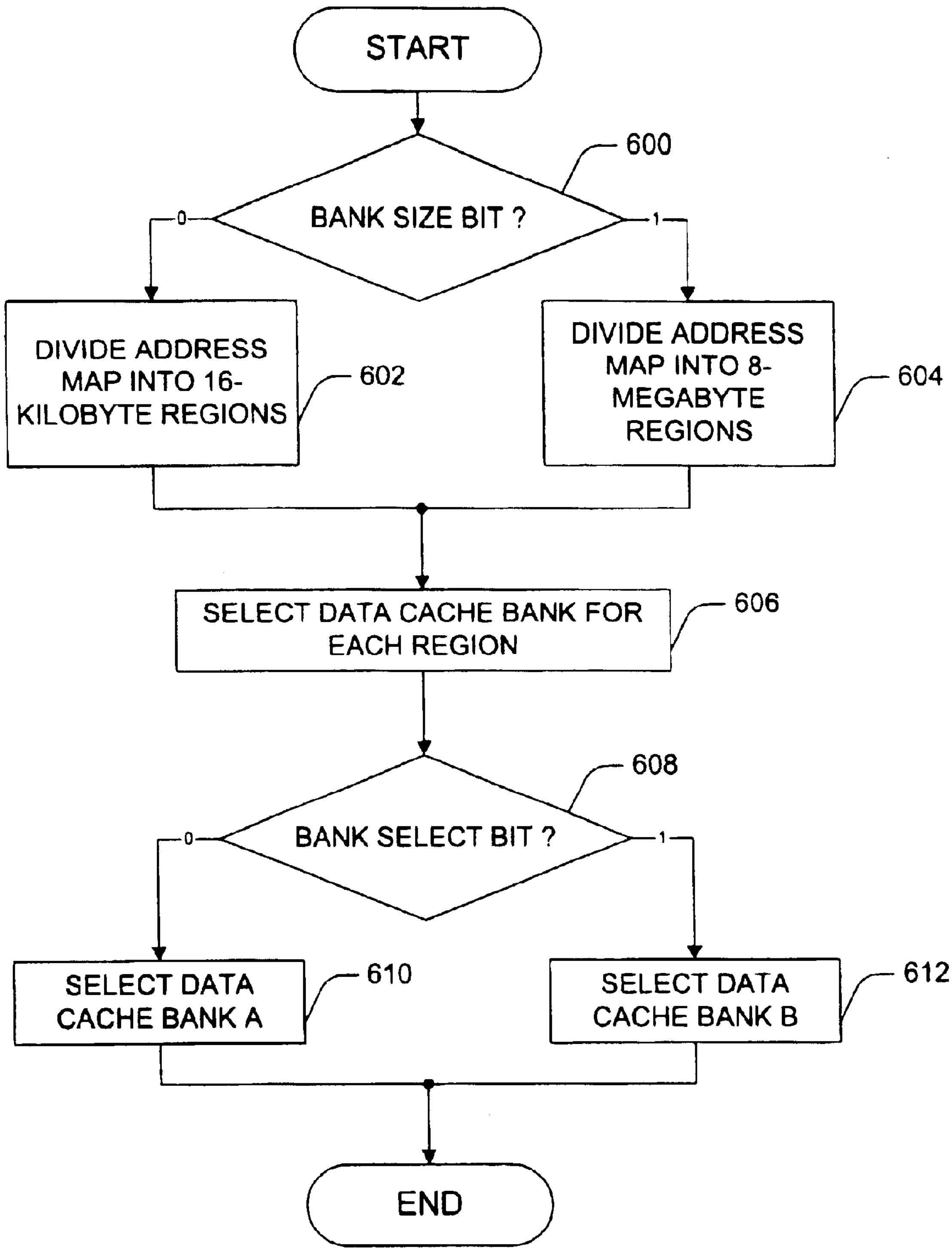

START
BANK SIZE BIT ?
600
0
1
DIVIDE ADDRESS MAP INTO 16-KILOBYTE REGIONS
602
DIVIDE ADDRESS MAP INTO 8-MEGABYTE REGIONS
604
SELECT DATA CACHE BANK FOR EACH REGION
606
BANK SELECT BIT ?
608
0
1
SELECT DATA CACHE BANK A
610
SELECT DATA CACHE BANK B
612
END

FIG. 6

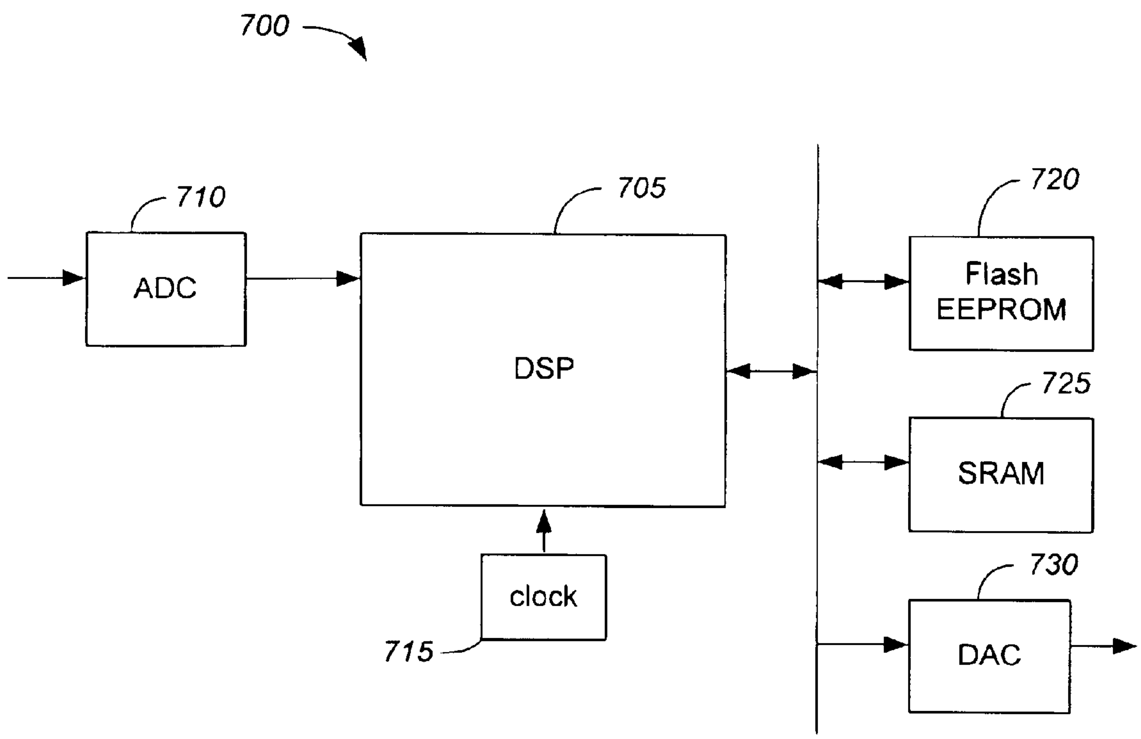

700
710
ADC
705
DSP
clock
715
720
Flash
EEPROM
725
SRAM
730
DAC

FIG. 7

MULTI-TIERED MEMORY BANK HAVING DIFFERENT DATA BUFFER SIZES WITH A PROGRAMMABLE BANK SELECT

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation application of and claims priority to U.S. application Ser. No. 09/541,114, filed on Mar. 31, 2000, now U.S. Pat. No. 6,606,684.

BACKGROUND

This disclosure generally relates to digital signal processing and other processing applications, and specifically to a programmable bank selection of banked cache architecture in such an application.

A digital signal processor (DSP) is a special purpose computer that is designed to optimize performance for digital signal processing and other applications. The applications include digital filters, image processing and speech recognition. The digital signal processing applications are often characterized by real-time operation, high interrupt rates and intensive numeric computations. In addition, the applications tend to be intensive in memory access operations, which may require the input and output of large quantities of data. Therefore, characteristics of digital signal processors may be quite different from those of general-purpose computers.

One approach that has been used in the architecture of digital signal processors to achieve high-speed numeric computation is the Harvard architecture. This architecture utilizes separate, independent program and data memories so that the two memories may be accessed simultaneously. The digital signal processor architecture permits an instruction and an operand to be fetched from memory in a single clock cycle. A modified Harvard architecture utilizes the program memory for storing both instructions and operands to achieve full memory utilization. Thus, the program and data memories are often interconnected with the core processor by separate program and data buses.

When both instructions and operands (data) are stored in the program memory, conflicts may arise in the fetching of instructions. Certain instruction types may require data fetches from the program memory. In the pipelined architecture that may be used in a digital signal processor, the data fetch required by an instruction of this type may conflict with a subsequent instruction fetch. Such conflicts have been overcome in prior art digital signal processors by providing an instruction cache. Instructions that conflict with data fetches are stored in the instruction cache and are fetched from the instruction cache on subsequent occurrences of the instruction during program execution.

Although the modified Harvard architecture used in conjunction with an instruction cache provides excellent performance, the need exists for further enhancements to the performance of digital signal processors. In particular, increased computation rates and enhanced computation performance of the memory system can provide advantages.

BRIEF DESCRIPTION OF THE DRAWINGS

Different aspects of the disclosure will be described in reference to the accompanying drawings wherein:

FIG. **1** is a block diagram of a digital signal processor (DSP) in accordance with one embodiment of the present invention;

FIG. **2** is a block diagram of a memory system containing two super-banks according to one embodiment of the present invention;

FIG. **3** is another embodiment of the memory system showing the mini-banks;

FIG. **4** shows a cache address map divided into contiguous memory regions of 16 kilobytes each according to one embodiment;

FIG. **5** shows a cache address map divided into contiguous memory regions of 8 megabytes each according to one embodiment;

FIG. **6** is a programmable bank selection process in accordance with one embodiment of the present invention; and

FIG. **7** is a block diagram of a system including a digital signal processor according to an embodiment.

DETAILED DESCRIPTION

A processor's memory system architecture can have a significant impact on the processor performance. For example, fast execution of multiply-and-accumulate operations requires fetching an instruction word and two data words from memory in a single instruction cycle. Current digital signal processors (DSP) use a variety of techniques to achieve this, including multi-ported memories, separate instruction and data memories, and instruction caches. To support multiple simultaneous memory accesses, digital signal processors use multiple on-chip buses and multi-ported memories.

Enhanced performance of the memory system can be accomplished using single-ported memory array having "multi-ported" behavior. Parallel accesses to multiple banks can be performed by providing configurable, fast static random access memory (SRAM) on chip. Alternatively, the memory system can be configured with caches, which provide a simple programming model.

A block diagram of a digital signal processor (DSP) **100** in accordance with one embodiment of the present disclosure is shown in FIG. **1**. The DSP is configured in a modified Harvard architecture. Principal components of the DSP **100** are a core processor **102**, an I/O processor **104**, a memory system **106** and an external port **108**. The core processor **102** performs the main computation and data processing functions of the DSP **100**. The I/O processor **104** controls external communications via external port **108**, one or more serial ports and one or more link ports.

The DSP **100** is configured as a single monolithic integrated circuit. In one embodiment, the memory system **106** implementation supports the SRAM-based model with two super-banks of 16 kilobits each for a total of 32 kilobits. These two super-banks of memory are accessed simultaneously in each cycle to support the core processor requirements. Alternatively, each of these super-banks can be configured as cache memory.

A first memory bus **120** interconnects the core processor **102**, I/O processor **104**, and memory system **106**. A second memory bus **122** likewise interconnects core processor **102**, I/O processor **104**, and memory system **106**. In some embodiments, the first memory bus **120** and the second memory bus **122** are configured as a data memory bus and a program memory bus, respectively. An external port (EP) bus **124** interconnects I/O processor **104** and external port **108**. The external port **108** connects the EP bus **124** to an external bus **126**. Each of the buses **120**, **122** includes a data bus and an address bus. Each of the buses includes multiple lines for parallel transfer of binary information.

The core processor **102** includes a data register file **130** connected to the first memory bus **120** and the second memory bus **122**. The data register file **130** is connected in parallel to a multiplier **132** and an arithmetic logic unit (ALU) **134**. The multiplier **132** and the ALU **134** perform single cycle instructions. The parallel configuration maximizes computational throughput. Single, multi-function instructions execute parallel ALU and multiplier operations.

The core processor **12** further includes a first data address generator (DAG**0**) **136**, a second data address generator (DAG**1**) **138** and a program sequencer **140**. A bus connect multiplexer **142** receives inputs from the first memory bus **120** and the second memory bus **122**. The multiplexer **142** supplies bus data to data address generators **136**, **138** and to the program sequencer **140**. The first data address generator **136** supplies addresses to the first memory bus **120**. The second data address generator **138** supplies addresses to the second memory bus **122**.

The core processor **102** further includes an instruction cache **144** connected to the program sequencer **140**. The instruction cache **102** fetches an instruction and two data values. The instruction cache **102** is selective in that only the instructions whose instruction fetches conflict with data accesses are cached.

For some embodiments, the DSP **100** utilizes an enhanced Harvard architecture in which the first memory bus **32** transfers data, and the second memory bus **34** transfers both instructions and data. With separate program and data memory buses and the on-chip instruction cache **144**, the core processor **102** can simultaneously fetch two operands (from memory banks **110**, **112**) and an instruction (from cache **144**), all in a single cycle.

The memory system **106**, illustrated in detail in FIG. **2**, preferably contains two super-banks of 16 kilobits each for a total of 32 kilobits. The super-banks A **200** and B **202** are accessed simultaneously in each cycle to support the core processor **102** requirements.

Each of these super-banks **200**, **202** can be configured as a SRAM and/or cache. By supporting both an SRAM and cache implementations together, the memory architecture provides flexibility for system designers. Configuring the memory as all cache helps the system designer by providing an easy programming model of the data cache for the rest of the code (e.g. operating system, micro-controller code, etc.). Configuring it as all SRAM provides predictability and performance for key digital signal processing applications. The hybrid version, e.g. half SRAM and half cache, allows mapping of critical data sets into the SRAM for predictability and performance, and mapping of the rest of the code into the cache to take advantage of the easy programming model with caches. Further, by providing SRAM behavior at the L1 memory, significant performance advantage can be achieved with low access latencies. In addition to the two super-banks, a 4-kilobit scratchpad SRAM **204** is provided as a user stack to speed up data switches.

In one embodiment, each of the data super-banks **200**, **202** is 16 kilobits in size and is further divided into four 4-kilobit mini-banks **300**, **302**, **304**, **306**. FIG. **3** shows a more detailed block diagram of the memory system **106**. In the illustrated embodiment, each mini-bank **300**, **302**, **304**, **306** is a two-way set associative cache and is configured as a single-ported memory array. By providing parallel accesses to eight different mini-banks **300**, **302**, **304**, **306** in the two super-banks A and B, a "multi-ported" memory behavior can be achieved. Multiplexers **308**, **310**, **312**, **314** selectively provide accesses of the mini-banks **300**, **302**, **304**, **306**, respectively. The selective accesses are provided to the core processor **316** and the system interface **318**, such as an I/O processor. However, since the configuration is not a true multi-port system, simultaneous accesses to a same mini-bank are not allowed. Thus, if two accesses are addressed to the same mini-bank, a conflict results. One of the accesses is delayed by one clock cycle.

For one particular embodiment, the first data address generator **322**, the second data address generator **324**, and the store buffer **320** provide addresses for two operands and a result, respectively.

The core processor **316** controls the configuration super-banks A and B of the memory system **106**. The configuration can be defined as described below in Table 1

TABLE 1

Memory Configuration	Super-bank A	Super-bank B
0	SRAM	SRAM
1	Reserved	Reserved
2	Cache	SRAM
3	Cache	Cache

The memory configurations **0** and **3** divide each super-bank into four mini-banks of all SRAM and all cache design, respectively. Each configuration provides either flexibility or ease of programming for the rest of the code. The memory configuration **2** supports hybrid design that allows mapping of critical data sets into the SRAM for predictability and performance, and mapping of the rest of the code into the cache to take advantage of the easy programming model with caches. When the SRAM mode is enable, the logical address and physical address are the same. The memory configuration **1** is reserved for a future configuration.

FIGS. **4** and **5** show examples of L1 cache memory organization. For the illustrated embodiments of the physical memory address map, bank selection is performed to allow parallel cache accesses of different buffer sizes. For example, FIG. **4** shows a 4-gigabyte cache address map divided into contiguous memory regions of 16 kilobytes each. The memory regions can be alternately mapped to one of two cache super-banks A and B. In another example of FIG. **5**, a cache address map is divided into contiguous memory regions of 8 megabytes each. For some embodiments, the cache address map is programmable to any practicable bank size. In addition, the bank size can be programmed dynamically so that the size can be modified in real-time according to specific implementations. The programmable selection has no effect unless both of the two cache super-banks A and B are configured as cache.

The organization of L1 cache memory allowing programmable bank size offer certain advantages over fixed bank size. Programming the memory into relatively small bank size offers advantage of increasing the chances that un-optimized code accesses both banks of cache. Large bank size favors applications with large data buffers, where a programmer needs to map large buffers into one bank for optimal performance.

FIG. **6** shows a programmable bank selection process in accordance with one embodiment of the present invention. At **600**, a bank size selection bit is queried to determine the cache memory bank size. If the bank size selection bit is zero, the address map is divided into contiguous memory regions of 16 kilobytes each at **602**. Otherwise, if the bank size selection bit is one, the address map is divided into memory regions of 8 megabytes each at **604**. At **606**, it is determined which data cache bank (i.e. A or B) is mapped to each region. This determination is made by using a bank select bit or by monitoring certain bits in the physical memory address. If the bank select bit is used at **608**, data cache bank A is selected at **610** if the bit is zero. Otherwise, data cache bank B is selected at **612** if the bit is one.

A truly multi-ported memory array can provide the bandwidth of two core processor accesses and a direct memory access (DMA) through such an interface as the system interface. However, the area penalty may be large because multi-porting of a memory array can more than double the physical area of the array. Furthermore, the cost of building a multi-ported array often increases exponentially. The memory architecture with multiple memory banks, as described above, can support parallel accesses with minimal hardware overhead. The arrays are single-ported, yet they can provide certain advantages of multi-port behavior, as long as the accesses are to different mini-banks.

The system environment can be optimized for maximum performance with minimal hardware. If DMA accesses are allowed into the cache, complex cache coherency issues are introduced that may result in control complexity and additional hardware. Thus, DMA accesses can be restricted only into the SRAM space. DMA accesses to the 4-kilobit scratchpad SRAM can also be restricted for simplicity.

Besides area advantage, multi-banking memory provides high access bandwidth, which is advantageous for digital signal processor performance. When in cache mode, a super-bank can support two core processor accesses in parallel with a fill or copyback transfer. When in SRAM mode, a super-bank can support dual core processor accesses in parallel with a DMA transfer. Further, power consumption can be reduced to a minimum by powering only the mini-banks that are needed by the accesses in a given cycle. At most, 3 out of 8 mini-banks are used per cycle.

Above described embodiments are for illustrative purposes only. Other embodiments and variations are possible. For example, even though the memory system has been described and illustrated in terms of having two different bank sizes and locations, the memory system can support having many different bank sizes and locations.

The DSP according to an embodiment of the present invention may be used in place of an ASIC in devices requiring digital processing. Some examples include digital video cameras, computers, cellular telephones, and personal digital assistants. For example, the DSP of according to one embodiment of the invention may be used in a mobile video communicator with Internet access. The DSP may perform the calculations necessary to process the video data.

FIG. **7** shows an exemplary system **700** which may include a DSP **705** according to an embodiment. The system may include an analog-to-digital converter (ADC) **710** to convert analog signals into digital signals to be operated on by the DSP. A clock **715** may be used to control the rate at which the DSP runs. An EEPROM (electrically erasable programmable read-only memory) **720** and SRAM **725** (static random access memory) may store instructions and data used by the DSP at runtime. A digital-to-analog converter (DAC) **730** may convert the digital signals to analog signals for output or display to a user of the system.

A number of embodiments have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. Accordingly, other embodiments are within the scope of the following claims.

What is claimed is:

1. A system comprising:

an SRAM memory device;

a clock; and

a digital signal processor coupled to the SPAN memory device and the clock, the digital signal processor including

a core processor;

a cache memory coupled to said core processor, said cache memory having a first block and a second block, where said first block and said second block are connected to said core processor in such a way as to allow substantially simultaneous data accesses for said core processor;

an address map including

a first region mapped to the first block, and

a second region mapped to the second block; and

a selector operative to

cache data having an address in said first region of the address map in the first block, and

cache data having an address in said second region of the address map in the second block,

wherein said first region and said second region are contiguous regions in the address map.

2. The system of claim **1**, wherein said first region of the address map and said second region of the address map have a selectable size which is large enough to allow mapping of buffers into a single region of said address map.

3. The system of claim **1**, wherein said simultaneous data accesses comprise accesses to both the first block and the second block in the same clock cycle.

4. The system of claim **1**, wherein the selector is operative to monitor a particular bit of said data address and determine whether to route the data to the first block or the second block based on a state of said bit.

5. The system of claim **4**, wherein the location of the particular bit in the data address corresponds to a size of said first and second regions of the address map.

6. The system of claim **1**, further comprising:

a third region in the address map, said third region being mapped to the first block,

wherein the second region and the third regions are contiguous regions in the address map.

* * * * *